

### Appendix 3

#### CALCULATION PROCESS

```
5  /*
   * Channel communicating object positions
   */ chap unsigned 17 position;

   /*
10  * Channel communicating segment information
   */
   chanout unsigned 9 segment;

   /*
15  * Channel communicating button information
   */
   chanin unsigned 2 buttons;

   /*
20  * Overall par
   */ par

       /*
       * Mass motion
25  */

       /*
       * Positions of each mass, 9+8 fixed point
       */
30  unsigned 17 p0, p1, p2, p3, p4, p5, p6, p7;
   /*
```

```
* Velocity of each mass, 9+8 fixed point
*/
int 17 v1, v2, v3, v4, v5, v6, v7; '
/*
5      * Accelerations of each mass, 9+8 fixed point
      */
      int 17 a1, a2, a3, a4, a5, a6, a7;
      /*
10     * Sutton status
      */
      unsigned 2 button status;
      /*
15     * Initial setup of positions
      */
      p0 = 65536;
      p1 = 65536;
      p2 = 65536;
      p3 = 65536;
      p4 = 65536;
20     p5 = 65536;
      p6 = 65536
      p7 =65536

25     /*
      * Forever
      */
      while (1)
      {
30
      /*
```

```

` * Send successive positions down position channel
    */
    send(position, p0);
    send(position, p1);
5    send(position, p1);
    send(position, p2);
    send(position, p2);
    send(position, p3);
    send (position, p3);
10    send(position, p4);
    send(position, p4);
    send(position, p5);
    send(position, p5);
    send(position, p6);
15    send(position, p6);
    send(position, p7);

    /*
    * Update positions according to velocities
    */
20    p1 += (unsigned 17)v1;
    p2 += (unsigned 17)v2;
    p3 += (unsigned 17)v3;
    p4 += (unsigned 17)v4;
25    p5 += (unsigned 17)v5;
    p6 += (unsigned 17)v6;
    p7 += (unsigned 17)v7;

    /*
30    * Update velocities according to accelerations
    */
```

```

v1 += a1 - (v1 » 6);
v2 += a2 - (v2 » 6);
v3 += a3 - (v3 » 6);
v4 += a4 - (v4 » 6);
5   v5 += a5 - (v5' » 6);
v6 += a6 - (v6 » 6);
v7 += a7 - (v7 » 6);

/*
10   * Set accelerations according to relative positions
    */
a1 = (int 17)(((p2 » 8) - (p1 » 8)) + ((p0 » 8) - (p1 » 8)));
a2 = (int 17)(((p3 » 8) - (p2 » 8)) + ((p1 » 8) - (p2 » 8)));
a3 = (int 17)!!((p4 » 8) - (p3 » 8)) + ((p2 » 8) - !p3 » 8)));
15   a4 = (int 17)(((p5 » 8) - (p4 » 8)) + ((p3 » 8) - (p4 » e> >);
a5 = (int 17)((!p6 » 8) - (p5 » 8)) + ((p4 » 8) - (p5 » 8)));
a6 = (int 17)(((p7 » 8) - (p6 » 8)) + ((p5 » 8) - (p6 » e> ) );
a7 = (int 17)((p6 » 8) - (p7 » 8));

20   /*
    * Get button information
    */
receive(buttons, button status);

25   /*
    * Fix top point according to buttons
    */ if (button status & 1)

p0 = 65536 - 16384;
30   )
else if (button status & 2)
```

```
(
    -
    p0 = 65536 + 16384;

else

5
    p0 = 65536;
    }
)
/*
10
    * nine drawing
    */
(
    /*
        * Positions of previous and next massess positions
15
        */
        unsigned 17 prev_pos, next pos, curr pos;
        /*
            * Which line of interpolation
            */
20
        unsigned char line;
        /*
            * Forever
            */
        while (1)
25
            (
                /*
                    * Receive previous mass position
                    */
                    receive (position, prev posy;
30
                    curr pos = prev pos;
                /*
```

```
* Read next mass position
*/
receive(position, next posy;
/*
5      * Do 64 lines of interpolation
*/
for (line = 0; line != 64; line++)
(
/*
10     * Send start position of segment
*/
send(segment, curr pos » 8);    /**width adjustment:17 along
                                channel of width 9 so takes bottom
                                9 bits*/
15
/*
    * Move by appropriate amount (1/64 total change)
    */
curr pos += (unsigned 17)(((int 17)next pos -
20                                (int 17)prev pos) » 6);
/*
    * Send end position of segment
    */
send(segment, curr pos » 8):
25
)
)
)
)
30
```

## DISPLAY PROCESS

```
/* standard includes */
#include "hammond.h"
5  #include "syncgen.h"
#include "stdlib.h"
#include "parallel.h"

/*
10  * Segment information channel */ chap segment;

/*
* Button information channel */
chan buttons:
15
/
* Include dash generated stuff */
#include "handelc.h"

20  /*
* Main program */
void main() (
/
* Scan positions
25  */ unsigned sx, sy;

/
* Vdeo output register
*/
30  unsigned l video;
```

```

        /*
        * Video output bus
        */

5   interface bus out() video out(Visible(sx, sy) ?
    (video ? (unsigned 12)Oxffff : 0) 0) with video spec;

    #ifndef SIMULATE
        /*
10   * Left button input bus
        */
        interface bus in (unsigned 1) button_left()
            with button white spec;

15   /*
        * Right button input bus
        */
        interface bus in(unsigned 1) button right()
            with button_black spec;

20   #endif

        /*
        *
        Overall par
25   */ par {
        /*
            * VGA sync generator
            */
            SyncGen(sx, sy, hsync pin, vsync pin);
30   /*
            *

```



Dash generated hardware

```

    */
hardware();
    /*
5      * Run-length decoder
      */
      {
    /*
10     * Segment start and end positions
      * /
      unsigned start, end;
    /*
      * Forever
      */
15   while (1)
      {
        while (sy != 448)
          /*
20           * Read segment information
          */
          segment ? start;
          segment ? end;
          /*
25           * Get in the right order
          */
          if (start > end)
            {
              par
                {
30
end = start;
```

```
start = end;.  
)
```

```
5          /*  
    * Make at least 1 pixel visible  
    */  
    if (start == end)  
        end++;  
10  
    /*  
        * Wait  
    */  
        while (sx != 0)  
            delay;  
15    /*  
        * Draw a scanline worth  
        * /  
        while (sx != 512)  
            if ((sx <- 9) >= start && (sx <- 9) < end)  
20  
                video = 1;  
                else  
                    video = 0;  
25  
            )  
        /*  
            * Communicate button status  
            */  
30    #ifdef SIMULATE  
        buttons ! 1;
```

```
#endif
```

 $\ast/$ 

delay;

)

**THE UNIVERSITY OF CHICAGO**